1. **Profiling: ArrayList Add-to-the-End**

**1.1 Source Code Being Profiled**

|  |
| --- |
| List<String> list = new ArrayList<String>();  for (int i = 0; i < n; i++) {  list.add("a string");  } |

**1.2 Profiling Results**

Total Runtime at Different Problem Size (n)

|  |  |
| --- | --- |
| **Problem Size (n)** | **Total Runtime (ms)** |
| 4000 |  |
| 8000 |  |
| 16000 |  |
| 32000 |  |
| 64000 |  |

Runtime vs. Problem Size on log-log Plot

Estimated SLOPE = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Therefore, the total runtime for the source code being profiled is \_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **QUADRATIC**),

so each *add (at the end)* operation of ArrayList is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **CONSTANT**) time.

1. **Profiling: LinkedList Add-to-the-End**

**2.1 Source Code Being Profiled**

|  |
| --- |
|  |

**2.2 Profiling Results**

Total Runtime at Different Problem Size (n)

|  |  |
| --- | --- |
| **Problem Size (n)** | **Total Runtime (ms)** |
| 4000 |  |
| 8000 |  |
| 16000 |  |
| 32000 |  |
| 64000 |  |

Runtime vs. Problem Size on log-log Plot

Estimated SLOPE = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Therefore, the total runtime for the source code being profiled is \_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **QUADRATIC**),

so each *add (at the end)* operation of ArrayList is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **CONSTANT**) time.

1. **Profiling: ArrayList Add-to-the-Beginning**

**3.1 Source Code Being Profiled**

|  |
| --- |
| List<String> list = new ArrayList<String>();  for (int i = 0; i < n; i++) {  list.add(0, "a string");  } |

**3.2 Profiling Results**

Total Runtime at Different Problem Size (n)

|  |  |
| --- | --- |
| **Problem Size (n)** | **Total Runtime (ms)** |
| 4000 |  |
| 8000 |  |
| 16000 |  |
| 32000 |  |
| 64000 |  |

Runtime vs. Problem Size on log-log Plot

Estimated SLOPE = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Therefore, the total runtime for the source code being profiled is \_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **QUADRATIC**),

so each *add (at the end)* operation of ArrayList is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **CONSTANT**) time.

1. **Profiling: LinkedList Add-to-the-Beginning**

**4.1 Source Code Being Profiled**

|  |
| --- |
|  |

**4.2 Profiling Results**

Total Runtime at Different Problem Size (n)

|  |  |
| --- | --- |
| **Problem Size (n)** | **Total Runtime (ms)** |
| 4000 |  |
| 8000 |  |
| 16000 |  |
| 32000 |  |
| 64000 |  |

Runtime vs. Problem Size on log-log Plot

Estimated SLOPE = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Therefore, the total runtime for the source code being profiled is \_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **QUADRATIC**),

so each *add (at the end)* operation of ArrayList is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (**LINEAR** / **CONSTANT**) time.